Measuring position bias in three LLM judges

I built a statistical evaluation harness for LLM systems, then used it to run a pre-registered study on the judges themselves: 300 prompts sampled from Arena-Hard-Auto at a pinned revision, two response pairs per prompt, every pair judged in both presentation orders by Claude Opus 5, Sonnet 5, and Haiku 4.5. 3,960 judge calls. The design, the falsification criteria, and the analysis were committed to the repository before the first API call.

Code, design document, and every validation table: github.com/Mnzrabusham/eval-harness

Judges prefer whatever is shown second

The cleanest measurement comes from the near-tie stratum, where both responses are independent samples from the same model on the same prompt. They are exchangeable by construction, and the data confirms it: the order-balanced content win rate is 0.509, 0.504, and 0.510 across the three judges, all straddling 0.5. There is no quality difference to detect.

Judge	β (near-tie)	Second position wins
Opus 5	–0.173 [–0.203, –0.145]	67%
Sonnet 5	–0.138 [–0.164, –0.113]	64%
Haiku 4.5	–0.148 [–0.180, –0.118]	65%

Given two responses of equivalent quality, all three judges pick the second one about two thirds of the time. This is recency, and it holds across a wide capability range. Pooled across both strata the effect is smaller, around –0.10, or roughly a 60/40 split.

The published literature finds both directions and reports that the direction is volatile across model families and tasks, so a within-family result like this one should not be read as universal.

A pre-registered check failed, and it matters

The design included a manipulation check: pair a Sonnet 5 response against a Haiku 4.5 response on the same prompt, and the stronger arm should win at least 65% of the time. All three judges failed it, at 0.536, 0.572, and 0.550.

Two explanations fit and this data cannot separate them. Either the judges discriminate poorly, or the two models genuinely produce near-equivalent 150-350 word answers on these prompts. The judges are highly consistent (ICC 0.80 to 0.94), so they are not guessing — but consistency is not discrimination, and a reliable judge can still be reliably indifferent.

What this costs: the stratified interpretation is void, and the pooled estimate mixes one clean stratum with one compromised one. What survives: β itself, which cancels content preference algebraically, and the near-tie result, which never depended on the manipulation working.

Reporting this is the point of pre-registering. A criterion that cannot fail is not a criterion.

A token ceiling does not just lose data, it biases what is left

The first run capped judge output at 64 tokens, sized for a one-line verdict. 11.9% of judgments were truncated mid-analysis and lost. Raising the ceiling to 256 changed almost nothing (12.2%). Raising it to 4,096 dropped the failure rate to 0.2%.

The interesting part is what the discarded judgments were. Retrying previously-successful calls at 4,096 agreed with the originals only 89.3% of the time, against a same-ceiling noise floor of 95.6% measured from replicate judging. The difference is –0.063 [–0.107, –0.021], and the equivalence test failed.

`max_tokens` is a stopping condition, not a generation parameter, so it does not change how the model samples. But a low ceiling censors on output length, and output length correlates with verdict category. The surviving sample is therefore biased with respect to the outcome. The direction is judge-specific: for Sonnet, short completions were nearly all ties; for Haiku the correlation ran the other way.

Truncation was also not distributed the way verbosity intuition suggests. At the 64-token ceiling: Sonnet 20.7%, Opus 13.8%, Haiku 1.4%.

Most evaluation harnesses cap judge output tightly to control cost.

Three things that broke quietly

Caching and independence are opposed by default. The replicate judge calls that exist to measure judge noise were computing the same cache key as their primaries, so they would have returned byte-identical completions and reported a noise variance of exactly zero. Caught in review, fixed for $2.57. This was the third cache-key collision of the same shape in the project.

Existence is not coverage. The runner checked that the execution plan file existed before spending money, which passed even when the file held only a stale plan for a different configuration. 543 plumbing errors, invisible until the join gate ran afterward.

A correlation that looked like a finding was a formatting artifact. Verdict category correlated strongly with completion length for every judge. For Sonnet the two were nearly collinear — because `VERDICT: TIE` is a shorter string to emit than a verdict plus its justification. Haiku showed the same correlation in the opposite direction, which rules out any single story about deliberation changing conclusions.

Limitations

One model family, so nothing here speaks to GPT or Gemini judges. One prompt source, skewed technical. Free-text verdict parsing rather than structured outputs, which the runner did not support. And the quality manipulation failed, so the clear-gap stratum measures less than it was designed to.

Monthir Ali — PhD, Computer Science. Contact: mnzrabusham@gmail.com